Amendments to the Specification

Please replace paragraph [0026] with the following amended paragraph:

26 A central processor 11 with main memory 12 connected to a main system address/data bus 13 links all the components of the hardware architecture. A secure trusted monitor program stored in the main memory 12 and executable by the central processor 11 controls all functions. A Boot Program 14 brings up the PAL 10 from a cold or warm start and runs test executables. The monitor program is stored in Flash Memory and System Software 16 where it can be updated as needed. Key session parameters have been stored from a previous use. Secure data such as encryption keys, financial data, owner name and pertinent data are stored in an encryption circuitry equipped random access memory 18. There is a micro video display controller 20 for a micro video display 22. Conventional voice analog circuitry 26 with speaker and microphone is utilized for voice digital conversion input/output 24. The monitor program stores specific voice samples for necessary biometric voice recognition. Additionally, the PAL 10 includes a barcode or optical reader subsystem 44 and optical scan assembly 46, magnetic stripe reader 34 utilizing a hardware interface 32 to the system bus 13, smart card reader 38 utilizing a hardware interface 36, remote ear piece 28, high-resolution touch screen display 50 with touch screen interface controller [[52]] 48, weight-measuring device 58 connected to the bus 13 by an analog-to-digital converter 56, a radio link controller 40 and radio subsystem 42 for high-speed secure short-range communication. Additionally, the PAL 10 includes a front panel 54 that includes keys, switches and indicators coupled to the bus 13 by a panel interface 52. As stated above, this listing of features is for illustrative purposes and not to limit the invention. Other features not listed are within the contemplation of the invention.

Please replace paragraph [0032] with the following amended paragraph:

32 An encode process, as shown in FIG. 3, shows how a vertical barcode or similar code can encode a WEB page display and/or behavior modifying rules. The encode process uses codes and checksums for data reliability. Returning to FIG. 2, the user will point a laser scan line 66 generated by the optical scan assembly [[42]] 46 at the top barcode 202 and drag the laser scan line 66 to the bottom barcode 203. When successful the purchasing aid logistic appliance 10 will beep one short high-pitched note. If not successful the purchasing aid logistic appliance 10 will beep one long low sounding beep. Checksums in the code will indicate a successful scan. Also, other types of signals representative of information can also be printed by the PAL 10.

Please replace paragraph [0033] with the following amended paragraph:

33 Returning to FIG. 3, the high-density barcode 60, which is designed to encode information required by the PAL 10 is scanned by a scanner [[102]] 102A, designed to extract the amount of information required to represent a WEB page. A decoder 103 decodes the scan using rules concerning how numbers are coded to represent information. These rules are coded into tables represented by numerals 105, [[106]] 106A,107 and 111. These tables interact with the decoder 103 to provide parsing information to a parser [[104]] 104A. The parser [[104]] 104A creates the software that will build the display. The parser [[104]] 104A can also identify rules and construct a Behavior Modifying Rules Table. The rules decode table 111 is a table that governs the rules of communications and commerce by the PAL 10. Specifically, the rules decode table 111 includes information specific to a unique merchant. When the PAL 10 is presented to the unique merchant, the rules decode table 111 will identify the merchant and terms and conditions of a sale. Information in the PAL 10 about other merchants will not be shared. The browser receives HTML or similar software from the parser [[104]] 104A and creates a display 109.

Please replace paragraph [0034] with the following amended paragraph:

34 The vertical barcode format, for example, does not contain a specific language, for example, Java or HTLM. Rather the vertical barcode will represent a sequence of numerical codes. Then from that, each suitable language will have a table from which a list of codes will generate a finite number of web page variations. The high-density barcode 60 identifies the table and represents codes indexed specifically for each language. The high-density barcode 60 uses the parser [[104]] 104A to construct the frame software needed by a display browser [[108]] 108A. This is a multi step process designed to make, for example, the vertical barcode format or a similar code independent of any browser language.

Please replace paragraph [0039] with the following amended paragraph:

39 The high-density barcode 60 or any similar print code contains a numerical table index code with data. The codes are cross-referenced to tables 105, [[106]] 106A,107,111, as shown in FIG. 3. The index points to a software instruction contained in the HTML decode table 105, which contains a series of code values 401, as illustrated in FIG. 5. These values contain significance to the parser [[104]] 104A. The HTML decode table 105 instructs the ~~parse 104~~ parser 104A how to interpret the value. For example, code 11 indicates the start of table data. The next value is one of eight possibilities corresponding to tables 105, [[106]] 106A,107,111.

Please replace paragraph [0040] with the following amended paragraph:

40 FIG. 6 is an example of the parsing process where the barcode 503 represents the numerical sequence 501. Embedded in numerical sequence 501 is code 11 12 05 which tells the

parser [[104]] 104A to select the HTML decode table 105 to interpret the remainder of the data and there are 5 code statements. Code 60 is a fictitious checksum value shown for example only. It will repeat at the barcode end. Code 22 indicates the start of table indices. Code 00 is the first entry in the HTML decode table 105. This first entry has no required fields. Index 04 is the first code statement to require a field. Code 31 tells the parser [[104]] 104A the next value is the number of characters in the first field. If there were a second field, code 31 would appear again after the 11th character code with a numerical representing the size of the next field. This process would continue until all fields were fully presented. Continuing with the example, 11 characters follow. They are coded in the range 31-56 which tells the parser how they to be interpreted as text characters. Finally code 21 appears again followed by 60. If the parser [[was]] were able to calculate the same checksum a short high pitch beep would indicate a successful scan and the resultant 502 would be put up on the display. Numerical codes are used to represent coding statements from any descriptive language that can build a display by way of browser instructions. The vertical barcode format does not require a specific language for example WAP, JAVA or HTML. Rather the frame software will be built from numeric codes. For example the HTML statement: *<p><table bgcolor="#000000" border=0 cellpadding=5 cellspacing=1 wIDth=468>* could be designated by the numeric code 43. The high-density barcode 60 is based on the premise that a web page can be constructed from a closed list of software statements. By way of the present example, ninety-nine statements could be developed and referenced by a two digit decimal code. By way of example, any practical number could be used for example 199. Data fields follow other codes as shown in FIG. 3. The parser [[104]] 104A can build the code statement and populate it with field data extracted from the printed code. At the conclusion of this process HTML or similar software is generated for a browser to generate a frame display.

Please replace paragraph [0041] with the following amended paragraph:

41 By way of example, the vertical barcode format can contain information about a merchant and sale terms and conditions. This information is displayed and may be entered into the rules table [[112]] 112A.

Please replace paragraph [0044] with the following amended paragraph:

44 The ~~secured~~ secure memory of the present invention, illustrated in FIG. 8, is included in a memory map 70, the most basic element of any computing device. The memory map 70 is organized into regions where specific tasks are performed. The regions are the physical address locations of a block of memory units 72. A memory unit 72 may be any number of bits but is usually a multiple of eight forming an eight-bit byte, sixteen-bit word, or thirty-two-bit double bit word. A single address location consists of several binary circuits, which must be decoded before the location can be opened for reading or writing. The central processor 11, illustrated in FIG. 1, sends out address signals on the address bus, which is processed by a memory address decoder. The decoder then selects which physical unit of memory is accessed. As shown in FIG. 8, an elementary memory map 70 in which the top of the map, RAM [[10]] 10A, represents an area where a program and data are stored, with logical methods of data transfer to the RAM [[10]] 10A, RAM input 13 and RAM output [[12]] 12A. Programs stored on permanent storage devices are accessed by software in the read only memory basic input/output operating system and put into RAM [[10]] 10A.

Please replace paragraph [0045] with the following amended paragraph:

45 The specific memory used to hold the video image to be displayed is represented by video memory 15. The difference between RAM [[10]] 10A and video memory 15 is the address

range, which is coded a read only memory basic input/output operating system (ROM BIOS) 25. When a program executing in RAM [[10]] 10A has data to write to the display, it calls the video out routine in ROM BIOS 25 and sends it data by way of the circuitry 17. This data is then displayed by the video generating circuitry connected to this memory (not shown), which is a well-understood process.

Please replace paragraph [0046] with the following amended paragraph:

46 A similar method is used to set encrypted RAM 20. Encrypted RAM [[20]] 20A has the same general properties of RAM [[10]] 10A in that memory can be read from and written to. It has two modes of operation; the first is secure and second is disabled. By way of read circuitry 23 and write circuitry [[22]] 22A encrypted RAM [[20]] 20A functions as ordinary memory when set to secure mode. When encrypted RAM [[20]] 20A is set disabled the data retrieved by way of read ~~circuity~~ circuitry 23 is not logical and therefore useless. Write circuitry [[22]] 22A does not function in a logical manner when disabled. The order of data is sometimes referred to in the literature as big Endian or little Endian. This is a reference to which byte of a multi-byte retrieval contains the most significant bit and which contains the least significant bit. Without knowing which causes the data to be improperly interpreted.

Please replace paragraph [0047] with the following amended paragraph:

47 FIG. 9 illustrates the use of encrypted RAM [[20]] 20A in the present invention in a three way verification process. In step 1 a smart card is inserted into the PAL 10 and the pin 30 is accessed. The user is prompted to enter a personal identification number (PIN) 31. The user's PIN number 31 is verified [[32]] 32S with the PIN 30 stored in the smart card. In step 2 a secure data hash 33 is compared to a hash [[34]] 34A stored in smart card or any convenient location. If

the comparison 35 is valid then the process continues to step 3, a verification 36S of a bond created in a previous session. From a previous session, a hash of the secure data in encrypted RAM was created and stored in two places. One place is the smart card and other is on the PAL 10. If both step 1 and step 2 are valid then the decision branch at step 4 is yes and encrypted RAM is unlocked 38S and made available. The user then performs one or more transactions, which may or may not change the data in encrypted RAM. Then a hash-creating algorithm located in ROM BIOS 25, see FIG. 8, runs and creates a new hash for the next session stored on the smart card data hash [[34]] 34A and on the PAL 10 for the next session. If steps 1 & 2 are not valid, then the data in encrypted RAM is destroyed.

Please replace paragraph [0048] with the following amended paragraph:

48 Figure 10 shows an alternative method of how encrypted RAM could be used in the PAL 10. First, two independent variables are generated 40S: the first variable 41 and the second variable [[42]] 42A. The first variable 41 and second variable [[42]] 42A combine in a process to generate a cipher key 43 that fits into a special address decoder [[44]] 44A. Ordinary RAM is attached to the special address decoder [[44]] 44A. Together, the special address decoder [[44]] 44A and RAM 45 create encrypted RAM [[46]] 46A.

Please replace paragraph [0050] with the following amended paragraph:

50 FIG. 11 is an alternative embodiment of encrypted RAM. A signal 49 starts and stops the encryption process. A portion of the RAM map [[20]] 70, as shown in FIG. 8, is set aside for secure memory. A special address decoder 55 and an address tracker [[56]] 56A generate the memory select lines using one of several possible mathematical formulae. This formula requires a random number be generated at the first time secure data is created then stored in address decoder

55. The random number encode or cipher key [[52]] 52A is added to the address to create an offset address from the correct location. The contents of a memory location are not encrypted but its address is intentionally misaligned by a random number incorporated into the address decoder. This random number is used whenever this secure address range is accessed in secure mode. Note that this technique applies to static RAM, Dynamic Ram and Flash RAM where an address decoder is required to generate select lines. For the purpose of this example shown in FIG. 11, a RAM segment [[54]] 54A is address encrypted. To keep the following example simple the memory RAM segment [[54]] 54A will be limited to 1024 bytes (hex B000-B3FF) of memory but any size is possible and larger is better.

Please replace paragraph [0051] with the following amended paragraph:

51 When the smart card is inserted into the PAL 10, the address decoder 55 reads and writes data in secure mode as normally directed by the central processor 11. In FIG. 11, a random number is generated by random number generation circuitry 51, hexadecimal F is used though any number is acceptable. The processor 11 now issues a write-to-memory command by calling for base address B000 50. The address decoder 55 receives B000 50 on the active address data bus from the central processor 11. The address decoder 55 also receives random number encode key hexadecimal F from the random number circuitry item 51. The address decoder 55 then computes memory select lines 53 as if the address were B00F. The central processor 11 writes a 4-byte variable so that 4 successive RAM 8 bit per byte locations are required. The next byte value from memory will be B001 bus address signals [[50]] 50A but is instead computed as select lines for B010 (B001+F) by address decoder 55. The central processor 11 thinks it is writing B000 through B003 four successive locations for this one variable. The four bytes are now stored in B00F, B010, B0011, B0012. When the smart card is removed, the address decoder 55 now gets a signal not to use the random number encode key [[52]] 52A generated by random number

generation circuitry 51. Now the central processor 11 attempts to read what has been written and issues 4 read commands by memory bus address signals [[50]] 50A beginning with location B000 through B003. The address decoder 55 functions normally but the data retrieved from these four locations B000-B003 is not coherent and therefore unintelligible. Next the smart card is again inserted using the previous example illustrated in FIG. 9 so that the address decoder 55 is now set to secure mode. The address decoder 55 now gets a signal to use the random number encode key. The four address locations B000-B003 are changed to B00F, B010, B0011, B0012 the exact same locations written to previously. The data is correctly retrieved. This example shows how data can be safely encrypted by intentionally misaligning the address. There is no record of the encode key stored in the RAM [[10]] 10A as shown in FIG. 9, the RAM [[10]] 10A where program and data are kept. Therefore the key cannot be retrieved by any method. Alternative embodiments include using –F instead of F as the encryption key, and binary compliment arithmetic could be used in place of simple addition. There are several valid methods other than strict interpretation of the above example capable of achieving the same purpose.

Please replace paragraph [0052] with the following amended paragraph:

52 Returning to FIG. 11, the central processor 11 keeps track of where data is stored using well-understood programming techniques. Using encrypted RAM, the central processor 11 cannot accurately track where data is physically stored. It depends on the address decoder 55 to correctly interpret the hexadecimal address. When the central processor 11 selects an address close to the end of encrypted RAM boundary, the address decoder 55 may set select lines 53 to go past the last boundary address. In the previous example that address is B3FF. The address select lines 53 will trap any address computed to be past the boundary. In the previous example if the central processor 11 set the address lines to B3F1 and the address decoder 55 was set to secure mode then the address decoder 55 would set the select lines for an address of B400 (B3F1

+ F), where the upper boundary is B3FF. The address tracker 56 senses the condition and sends a signal [[58]] 58A, ranging between 0 and E, generated by the address tracker [[56]] 56A to the address decoder 55. This signal causes the address decoder 55 to reset the select lines 53 as if base address B000 were being decoded. The decode key 57 is generated by an encode cipher key [[52]] 52A for use by the address tracker [[56]] 56A. The decode key 57 is used to compute the value of the signal [[58]] 58A. No address is used twice and no valid memory contents are overwritten and the central processor 11 cannot determine the equivalent address used.

Please replace paragraph [0053] with the following amended paragraph:

53 The encode cipher key [[52]] 52A can be changed from time to time to maintain an element of randomness. The random encode cipher key [[52]] 52A may be generated once per active session. An active session is defined as a continuous RAM [[54]] 54A power cycle. As long as the RAM [[54]] 54A is active the encode cipher key 52 is not lost. If power to the RAM [[54]] 54A is lost, then the encode cipher key [[52]] 52A is likewise lost and data lost with it as well. There are other possibilities for setting rules regarding the generation of an encode cipher key [[52]] 52A. For example a key [[52]] 52A could be generated once per secure session and erased when the secure session is ended. In this case the data will be lost unless steps are taken to offload the encrypted RAM data to an alternate location. If the data were offloaded onto temporary storage then the encode key could be changed periodically and the secure data reloaded back to encrypted RAM. This would be better for security if the encode key were periodically changed. This technique makes it highly unlikely an external spoof can be used to strobe and read out secure memory.

Please replace paragraph [0055] with the following amended paragraph:

55 A radio subsystem [[32]] 42 (FIG. 1) utilizes two conventional types of antennae, as illustrated in FIG. 11a, simultaneously. These two antennas, one a forward directional antenna 41a and the other an omni directional antenna 43a are used by the protocol to affect the RF link. The forward directional antenna 41a includes conventional components, such as, a signal absorbing material 41b, reflecting cone shaped director 41c, and a directional element 41d. The omni directional antenna 43a includes conventional dual back-to-back hemispherical coverage antennas 43b. By managing the power, a link can be created within the confines of an aisle shelf area (to be discussed below). The omni directional antenna is used to communicate with the PAL 10 when it is not within the confines of the aisle shelf area. Both antennas are operated from their respective antenna controllers [[41]] 41e, [[43]] 43c, and connected to the processor 11 through the radio link controller 40. The advantages of two antennas are diversity and multi channel link control giving the merchant computer the ability to manage large numbers of simultaneous users.

Please replace paragraph [0059] with the following amended paragraph:

59 After the shopping list file has been uploaded, the merchant computer 66, see FIG. 7, returns pertinent data back to the PAL 10 while the PAL 10 is still traversing the doorway area [[50]] 150, see FIG. 12. A numerical ID value is assigned to each PAL 10 while it is in the facility. If desired, a customer may set the PAL 10 to provide customer identification through the use of a trusted surrogate ID that only has meaning to the merchant. If this information were to be intercepted it would have no value without the merchant computer's database. Likewise a customer may set the PAL 10 to deny customer private identification. On the other hand, the customer may set PAL 10 to accept a merchant's database pointer value, which becomes the customer's In-Store ID and returns this pointer value when communicating with any of the link

methods. The barker beacon 142 will identify the merchant, address date and time and next channel assignment. These values will be sufficient for the PAL 10 to retrieve the merchant's database pointer set from a previous time and uplink it along with the item list. Likewise the customer may choose not to send the database pointer but instead use a unique generic ID in place of the specific ID. Up linking the database pointer differentiates a patron user from a public user. The merchant will set customer treatment rules accordingly so that the merchant computer 66 sends the correct information to each user. For example the merchant may wish to exchange a personalized greeting for each patron and a general greeting for a public customer.

Please replace paragraph [0077] with the following amended paragraph:

77 Finally, an additional feature of the PAL 10 of this invention, as illustrated in FIG. 16, shows how PAL 10 can measure weight. Weight is a key parameter for measuring out quantity to determine price. The weight measuring device 58 includes a strain gauge 312 mounted on a shaft 314. The shaft 314 has a fixed end 315A and a free end 315B. The fixed end 315A is fixedly attached to the PAL and the free end 315B is rotatably attached to the PAL 10. A spring 320 is fixedly attached to the PAL 10 and the shaft free end 315B. A line 316 is wound on a pulley 318 having a slip knot mechanism [[320]] 320A on the free end 321 of the line 316. The slip knot mechanism [[320]] 320A allows the line to form an adjustable loop. The loop can be cinched to hold material without bottom support while the PAL 10 computes the weight of an object purchased. The pulley 318 is fixedly attached to the shaft 314. The line 310 is cinched tight to hold the object. The PAL 10 then determines the weight based on the strain measured by the strain gage 312 as the shaft 316 torsional deflects under the load of the object. The user then can add price input to determine the total price. The PAL 10 is equipped with a method for accurately measuring the weight of a small amount of mass. A pulley, shaft and springs are used to make a self-retracting mechanism for storing a strong lightweight braided line. The line material

is chosen so that it will not stretch over time or distort in any way when used within design limits. A line release mechanism 322 on the PAL 10 handle releases a lock (not shown) that allows the line 316 to be extended from the PAL 10. The strain gauge 312 is connected to the analog-to-digital converter 56, thereby linking the weight measuring device 58 to the processor 11, as illustrated in FIG. 1. The strain gauge deflections are transmitted to the central processor 11 for conversion into weight and calculating purchase price based on the cost per unit weight inputted by an input device, such as an optical scanner or keyboard.